ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Bachelor Thesis
in
Computer Science

Tuples-DMM: A retrieval-enhanced concept-driven Guided Decoding Algorithm

Dimosthenis Plavos

Supervisor: Assistant Prof. John Pavlopoulos
Department of Informatics
Athens University of Economics and Business

December 2024

Dimosthenis Plavos

Tuples-DMM: A retrieval-enhanced concept-driven Guided Decoding Algorithm

December 2024

Supervisor: Assistant Prof. John Pavlopoulos

Athens University of Economics and Business

School of Information Sciences and Technology

Department of Informatics

Information Processing Laboratory, Natural Language Processing Group

Athens, Greece

Abstract

Biomedical image captioning is an evolving task in the field of Artificial Intelligence that involves the automatic generation of descriptive captions for medical images. It is driven by the advancements in imaging technologies and the increasing volume of patients that have resulted in a large number of radiological images in healthcare facilities worldwide. Analyzing these images demands a great amount of time from clinicians, which makes the automation of this procedure a time-saving process. The automatically created captions can also serve as tools to guide the diagnosis process or confirm the findings of the clinicians. This thesis focuses on Diagnostic Captioning (DC), which refers to the generation of textual descriptions aimed at identifying and conveying diagnostic information from medical images. For its implementation, it utilizes the ImageCLEFmedical 2023 dataset. The proposed Tuples-DMM method builds upon the DMM (Distance from Median Maximum) approach, which is a key concepts-driven Guided Decoding method introduced by Kal et al. [Kal+24]. DMM method generates captions by explicitly or implicitly incorporating the tags associated with a medical image, based on how these tags are represented in its training captions. Tuples-DMM method and its modifications integrate strategies to retrieve the most relevant training data and modify the DMM algorithm. The goal is to enhance guided generation by minimizing the influence of captions that represent unrelated contexts and focusing on contextually relevant training data to achieve more accurate and meaningful captions.

Περίληψη

Η αυτόματη περιγραφή ιατρικών εικόνων αποτελεί μια εξελισσόμενη διαδικασία στον τομέα της Τεχνητής Νοημοσύνης που περιλαμβάνει την αυτόματη παραγωγή περιγραφικών λεζαντών για τέτοιες εικόνες. Ενισχύεται από τις προόδους στις τεχνολογίες απεικόνισης και τον αυξανόμενο αριθμό ασθενών, τα οποία έχουν οδηγήσει στη δημιουργία ενός μεγάλου αριθμού ακτινολογικών εικόνων στις μονάδες υγειονομικής περίθαλψης παγκοσμίως. Η ανάλυση αυτών των εικόνων απαιτεί σημαντική ποσότητα χρόνου από τους κλινικούς ιατρούς, γεγονός που καθιστά την αυτοματοποίηση αυτής της διαδικασίας ένα μέσο εξοικονόμησης χρόνου. Οι αυτόματα δημιουργούμενες λεζάντες μπορούν επίσης να χρησιμεύσουν ως εργαλεία για την καθοδήγηση της διαγνωστικής διαδικασίας ή την επιβεβαίωση των ευρημάτων των κλινικών ιατρών. Η πτυχιακή αυτή εργασία επικεντρώνεται στην Παραγωγή Διαγνωστικής Περιγραφής (Diagnostic Captioning), η οποία αναφέρεται στη δημιουργία κειμενικών περιγραφών με στόχο την αναγνώριση και μετάδοση διαγνωστικών πληροφοριών από ιατρικές εικόνες. Για την υλοποίησή της, χρησιμοποιεί το σύνολο δεδομένων ImageCLEFmedical 2023. Η προτεινόμενη μέθοδος Tuples-DMM βασίζεται στη μέθοδο DMM (Distance from Median Maximum), που αποτελεί μια μεθοδολογία Καθοδηγούμενης Αποκωδικοποίησης βασισμένη σε "κεντρικές έννοιες" και παρουσιάστηκε από τον Kaliosis και άλλους [Kal+24]. Η μέθοδος DMM δημιουργεί περιγραφές ενσωματώνοντας ρητά ή άρρητα τις έννοιες που σχετίζονται με μια ιατρική εικόνα, σύμφωνα με τον τρόπο που αυτές οι έννοιες εκπροσωπούνται στα παραδείγματα εκπαίδευσης. Η μέθοδος Tuples-DMM και οι τροποποιήσεις της στοχεύουν στην ανάκτηση των πιο σχετικών δεδομένων εκπαίδευσης και την τροποποίηση του αλγορίθμου DMM. Ο στόχος είναι η βελτίωση της καθοδηγούμενης δημιουργίας μέσω της αποφυγής της επιρροής από δεδομένα εκπαίδευσης που αντιπροσωπεύουν άσχετα νοηματικά θέματα και της εστίασης σε σχετικά νοηματικά δεδομένα εκπαίδευσης, προκειμένου να επιτευχθούν πιο ακριβείς και νοηματικά ουσιαστικές περιγραφές.

Acknowledgements

First of all, I would like to sincerely thank my supervisor, John Pavlopoulos, and Professor Ion Androutsopoulos for their guidance throughout the implementation of this thesis and for sharing their knowledge and advice. Their mentorship made me realize how research teams work, conduct research, and deliver impactful results, and helped me gain a deeper understanding of the field.

Moreover, I would like to express my gratitude to PhD candidate P. Kaliosis for guiding me through the entire process. His thesis' method served as a baseline for my work, and I am especially thankful for our discussions and his advice on academic and research concepts.

I would also like thank the other members of the AUEB NLP research group, including PhD candidates G. Moschovis and F. Charalampakos, as well as M. Samprovalaki and A. Chatzipapadopoulou, for their invaluable support with technical issues, helpful advice on my thesis, and for the offline conversations throughout the whole procedure.

The cooperative environment of the research group, to which each member contributed, made the entire procedure much more enjoyable and provided me with the opportunity to deepen my knowledge and feel genuinely supported.

Contents

1 Introduction

In recent years, medical imaging has become a very challenging field. It requires the ability to visualize the internal structure of the human body and depict it in images of specific forms, such as X-rays, Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) scans. These images are now used routinely by radiologists to draw conclusions about the medical condition of patients based on their visual characteristics. As these systems become more sophisticated, the complexity of their data increases dramatically. Although their expertise is focused on making accurate assessments, radiologists face the challenging task of examining large sets of images, which can be time-consuming, mentally demanding, and susceptible to human error.

To address these challenges, researchers and professionals have begun to explore AI-based solutions designed to help radiologists reduce workload while maintaining the quality of diagnoses. The goal is to create a descriptive and accurate caption that highlights the most important image characteristics and serves as a guide tool for clinicians during diagnosis. One promising approach involves Diagnostic Captioning systems, tools that combine Computer Vision (CV) with Natural Language Processing (NLP) to generate concise, medically accurate reports directly from radiological images. This thesis investigates DC systems regarding the ImageCLEF competition, which has two main tasks, Concept Detection and Caption Prediction. The first sub-task involves understanding the key features of a medical image and extracting key tags, which are keywords or phrases that describe the most important characteristics of the image and the medical conditions depicted. The second sub-task is the caption generation task, which aims to create diagnostic captions. In addition to the integration of these tags, there are also other rules to be followed, such as lexical constraints, depending on the methods and the language model used during the text generation procedure.

This thesis focuses on combining advanced decoding methods with retrieval-based techniques to improve diagnostic captioning. Expanding on the work of Kaliosis et al. [Kal+24], who employed guided decoding using a custom variant of the beam search algorithm to generate medical text, this approach utilizes a multimodal diagnostic system to produce diagnoses. Specifically, it uses the InstructBLIP [Dai+23] Vision-Language Model (VLM). VLMs are deep learning models capable of generating open-ended text or textual responses conditioned on a visual input. In that case, InstructBLIP uses a textual prompt and a specific image associated with medical tags. These tags -otherwise called concepts- are words or phrases that represent important characteristics of the image. With the image

and tags as input, the model runs the modified beam search algorithm, which explores multiple possible sequences of words at each step and selects the next word according to a scoring mechanism. These sequences that are explored simultaneously represent the beam and a beam score is computed for each one of them. These scores, which eventually define how the generated caption is constructed, are calculated based on the training captions associated with the medical tags of the image. By incorporating retrieval techniques directly into the beam search computation, the approach reduces the sample space of the training captions considered, thus focusing on more contextually important information to influence the decoding. The desired result is to produce meaningful and accurate captions that allow radiologists to quickly identify essential patterns and insights while maintaining a high level of clinical accuracy.

1.1 Thesis Structure

Chapter 2: Background & Related Work

In this chapter, I present background information and an overview of related work in the areas of generic and biomedical image captioning, guided decoding, and the co-occurrence of terms in information retrieval.

Chapter 3: Implemented Methods & Systems

Chapter 3 introduces the baseline method proposed by Kaliosis et al. and describes the newly implemented methods of this thesis.

Chapter 4: Data

This chapter provides an overview of the dataset used for this task.

Chapter 5: Experiments & Results

Chapter 5 presents and analyzes the experimental results obtained from the methods described in Chapter 3.

Chapter 6: Conclusions & Future Work

Chapter 6 summarizes the main conclusions drawn from this research and discusses suggestions for future work in the field.

2 Background and Related Work

2.1 Image Captioning

Image captioning, the task of generating descriptive text for a given image, has become a key area of research in Artificial Intelligence (AI) due to its wide range of applications. This task involves not only identifying the key objects within an image, but also understanding the attributes of the objects and their relationships to each other. The aim is to eventually generate a coherent description that is both grammatically correct and contextually accurate [Vin+17].

Traditionally, image captioning relied on hand-crafted features and classical machine learning techniques [SP23]. Methods such as Local Binary Patterns (LBPs) [OPM02], Global Image Descriptor (GIST) [OT01], Scale-Invariant Feature Transform (SIFT) [Low99], and Histograms of Oriented Gradients (HOGs) [DT05] were common for extracting image features, which were then fed into classifiers such as Support Vector Machines (SVM) to generate captions [Hos+19]. However, these approaches were limited by their dependence on task-specific features, making it difficult to adapt to diverse datasets or complex visual material.

With the evolution of deep learning, there has been a shift in how image captioning is approached, focusing on more advanced and automated approaches. Deep learning techniques, particularly Convolutional Neural Networks (CNNs) [Ste+23] [KK20] for feature extraction and Recurrent Neural Networks (RNNs) [Wan+19] or Transformers [WXS22] for text generation, have become the most prominent methods [SP23]. Unlike traditional methods, these models have the advantage of being able to learn and generalize features from large and diverse datasets. Therefore, they can capture the intricate details and relationships within real-world images, making them more adaptable and accurate at generating text compared to traditional methods, such as those mentioned above.

2.1.1 Medical Image Captioning

Medical image captioning, a specialized branch within the broader field of image captioning, focuses on the automatic generation of textual descriptions for medical images, such as X-Rays, MRIs, and CT scans [Par+21a]. By combining techniques in Computer Vision

and NLP, this technology aims to assist radiologists and other medical professionals to interpret and document these images more effectively, by automating parts of the diagnosis generation process. Otherwise, they they can serve as a tool to validate professionals' diagnoses. Given the vast number of images that clinicians need to interpret daily, the development of reliable and efficient medical image captioning systems has become an important research area [KPA19].

Unlike general images, medical imagery faces unique challenges, such as the demonstration of specific medical conditions and the usage of complex medical terminology. These challenges make the development of effective medical image captioning methods both highly valuable and significantly more complex. In order to be addressed, they require a deeper understanding of image captioning techniques and how they can be adapted to meet the demands of the medical domain. Over the years, several approaches to image captioning have been proposed, each using different strategies to generate descriptions. These methods can be broadly classified into three main types: template-based, retrieval-based and LM-based methods [Hos+19].

2.1.2 Image Captioning Methods

Template-Based Methods

Template-based methods for image captioning rely on predefined sentence structures with blank slots which are filled using detected objects, attributes, and actions found within an image [DCI19]. Although these methods are capable of generating grammatically correct captions, they struggle with producing captions that vary in length or can adapt to complex image content due to their lack of flexibility.

According to Liu et al. [LXW19], the process of template-based image captioning can be divided into two key subtasks. The first involves object detection and classification using Computer Vision techniques. The second subtask translates the detected objects into sentences using language models (LMs), taking into account the objects' attributes and their relationships with the environment. For example, when using a template-based method, visual object detectors recognize and analyze objects in images, identifying a set of words commonly used in captions. These words are then inserted into a predefined template, which is supplemented with additional phrases to form a complete sentence. One advantage of template-based methods is the ability to independently debug each module in the process, such as the object detection or sentence generation components.

Early methods, such as those by Farhadi et al. [Far+10], used deformable part-based object detectors, modeled after Felzenszwalb et al.'s [FMR08] Deformable Part Models (DPMs) detectors to identify triplets of scene elements (subject, object, and action), which were

then converted into a sentence using a predefined template. For example, the triplet "man, riding, horse" can generate a sentence such as "A man is riding a horse." Later works built on this foundation.

Retrieval-Based Methods

Retrieval-augmented text generation combines traditional retrieval techniques with deep learning approaches to enhance the performance of various NLP tasks. In contrast to traditional models, which rely solely on learned parameters to generate text, a retrieval-based model leverages external memory to retrieve relevant information and inject it into the generation process, reducing the model's dependence on memorization. Therefore, retrieval-based methods have gained significant attention due to the scaling of model parameters and offer improvements in performance without significantly increasing computational costs [Hos+19] [Li+22].

As Liu et al. [LXW19] explain, when applied to image captioning tasks, these methods use a combination of retrieval mechanisms and LMs to generate more contextually accurate and semantically rich descriptions. The model finds images in the training dataset that are visually similar to the query image and then retrieves their associated captions to generate new captions. In general text generation tasks, the retrieved information could be similar captions or textual elements from an external memory based on textual similarities.

Diving into retrieval-based methods, Lewis et al. [Lew+21] describe Retrieval-Augmented Generation (RAG) models, which offer a powerful approach by merging parametric memory (static knowledge stored within model parameters) with non-parametric memory (external sources) to improve text generation tasks. Traditional pre-trained models, such as BART or T5, although successful in storing a significant amount of knowledge within their parameters, struggle with dynamically updating information and providing provenance for their outputs. However, RAG models integrate a retrieval mechanism, allowing them to access explicit seq2seq model serves as the parametric memory, while the non-parametric memory is represented by a dense vector index of external knowledge, such as Wikipedia, which is accessed through a pre-trained neural retriever. According to Lewis et al., there are two main variants of RAG used to generate text, the RAG-Sequence model and the RAG-Token model. The RAG-Sequence model retrieves the top-K documents to guide the generation of the output sequence and the RAG-Token model retrieves different documents for each token in the generated sequence, allowing for more diverse and contextually appropriate information to be incorporated throughout the generation. The RAG text generator is typically a pre-trained BART model, which generates text conditioned on both the input and the retrieved documents [Kar+20].

In addition to RAG architectures, there are also specific mechanisms that help in the text generation process, using the retrieved captions. The method outlined by Hossain et al.

[Hos+19] relies on kNN searches in the visual feature space, where the relevance of the retrieved captions is determined based on the similarity between the input image's visual embedding and those stored in the memory.

Some other retrieval-based methods are FiDO (Fusion-in-Decoder Optimized), REALM (Retrieval-Augmented Language Model Pre-Training) and KNN-LM. FiDO [Jon+23] directly integrates the retrieved documents into the Transformer model decoder. Unlike approaches that use retrieved documents as separate input or preprocess them, FiDO treats the retrieved documents as part of the decoding process itself. REALM [Guu+20] dynamically retrieves text from external documents during pre-training and fine-tuning. The KNN-LM method [Kha+20] compares, during inference, the hidden states of the current input with the stored embeddings of previous text the model has processed. By finding the best matches, it retrieves the most relevant tokens. Due to its ability to leverage large amounts of stored data, it is effective in predicting rare or long-tail sequences.

LM-Based Methods

The task of medical image captioning has evolved from earlier methods that relied primarily on rule-based or template-based approaches to more sophisticated deep learning based models [SJS22]. As explained in the previous sections, retrieval-based and template-based techniques offer limited flexibility and generalization, particularly in handling the vast diversity and complexity of medical images. Deep learning models have become more advanced, particularly with the introduction of CNNs, RNNs and Transformers. Therefore, recent research has focused on adapting and improving deep learning architectures to address the specific needs of medical image captioning. These methods aim to generate entirely new captions for each image by first analyzing the visual content and then using LMs to produce the corresponding text. A prevalent approach involves the use of an encoder-decoder framework [BI24]. The encoder, typically a CNN, extracts feature representations from the medical image, while the decoder, often an RNN or Transformer-based model, generates the corresponding textual description. At the same time, the introduction of attention mechanisms[Xu+15] has allowed models to focus on specific parts of an image while generating each word in a caption, thus improving the accuracy and relevance of the generated descriptions. An example of an image captioning task using deep learning methods is shown in Figure 2.1, derived from the work of Xu et al. [Xu+16]. In the next parts, RNNs, Transformers and attention mechanisms will be analyzed.

RNN-based Models

According to Sutskever et al. [SMH11] and Ghandi et al.[GPM23], RNNs are a powerful model for tasks involving sequential data, such as text generation, due to their ability to maintain context and process information one element at a time. RNNs are especially effective because they use their internal memory to "remember" past computations and

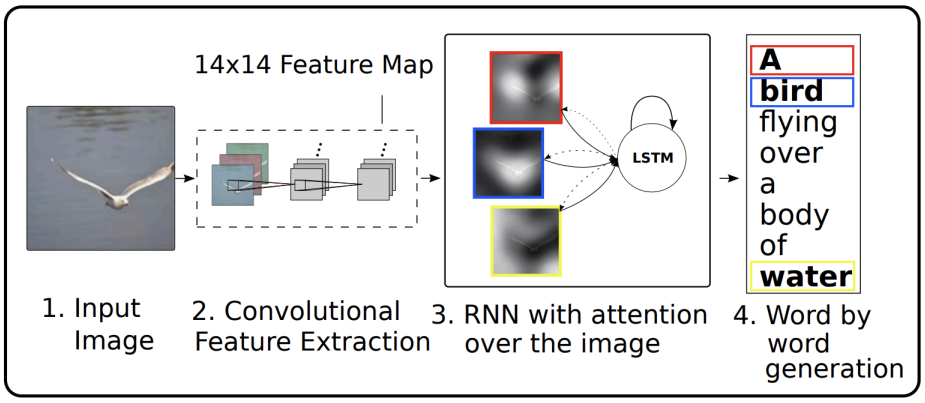

Fig. 2.1: The figure shows an example of image captioning, where a model generates a caption for an image by focusing on different parts of the image for each word. Image features are extracted using a CNN, and then an RNN with an attention mechanism processes these features to generate the caption word by word. The attention mechanism helps the model focus on the most relevant image areas at each step. The figure was created by Xu et al. [Xu+16].

apply this information to influence the output for the current input. This feedback loop enables the model to learn dependencies across sequences, making it capable of generating coherent, contextually relevant text that follows syntactic and semantic rules.

In text generation tasks, the sequence-to-sequence (seq2seq) learning framework is commonly used [Jin+20] [KB13]. Seq2seq models consist of an encoder and a decoder and are designed to process input sequences and generate corresponding output sequences. Shin et al. [Shi+16] were the first to apply the encoder-decoder approach in medical image captioning, using CNNs to encode medical images and RNNs to generate textual descriptions. This method demonstrated promising results, particularly when the CNN was trained in specific medical categories.

However, traditional RNNs face challenges, such as the vanishing gradient problem, which makes it difficult to learn long-term dependencies in sequences. This issue arises because the influence of earlier inputs in a sequence may diminish as more inputs are processed, leading to the loss of important information over time. To address this, improved versions of RNNs, such as Long Short-Term Memory (LSTM) networks [Hoc97] and Gated Recurrent Units (GRUs) [Cho+14] were developed. These models use “gates” to regulate the flow of information, allowing them to better retain relevant data throughout the sequence. This improvement in handling long-term dependencies has made LSTMs and GRUs the preferred choice in many text-generation tasks, particularly those requiring memory of longer sequences. Suresh et al.[SJS22] experimented with different CNN architectures, such as ResNet [He+16], and decoding strategies, such as greedy search and beam search. The findings indicate that certain combinations, particularly the ResNet-101 encoder paired with LSTM or GRU decoders, outperform traditional CNN-RNN models, especially when using advanced inject models and decoding techniques.

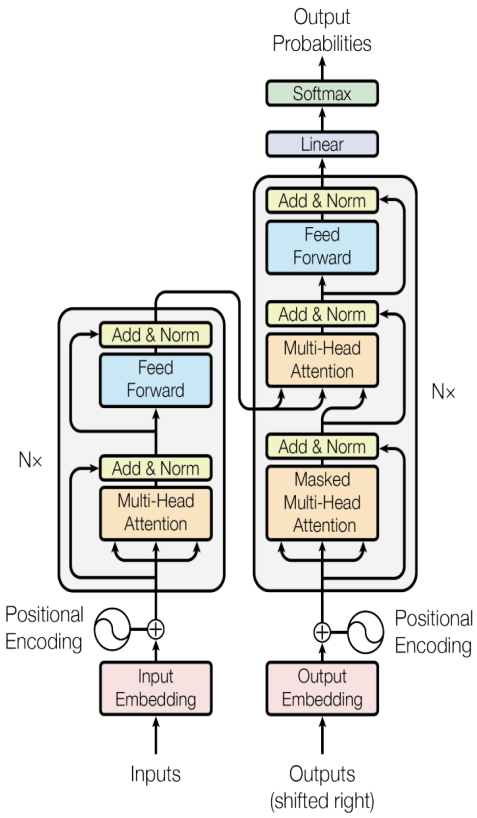

Fig. 2.2: The Transformer model features an encoder-decoder structure. The encoder processes input embeddings with positional encodings through layers of multi-head self-attention and feed-forward networks, each followed by residual connections and normalization. The decoder includes a masked self-attention mechanism to ensure outputs are generated step-by-step and integrates information from the encoder using multi-head attention. The final layer produces output probabilities via a linear transformation and softmax function. The figure was created by Vaswani et al. [Vas17].

Transformer-based Models

Vaswani and Polosukhin [Vas17] designed and implemented the first Transformer models, as shown in Figure 2.2. These models revolutionized previous architectures, such as RNNs and LSTM, which struggled with vanishing gradients and parallelization due to their sequential nature. The key innovation of Transformers lies in their ability to capture long-range dependencies in data using attention mechanisms, which allows them to handle longer sequences more efficiently. Transformers eliminate recurrence and convolutions, relying solely on attention to process data in parallel, which enables faster training on larger datasets. The success of Transformer models in NLP has led to their adoption in medical image captioning. As Topal et al. [TBH21] propose, Transformers have enabled models, such as GPT (Generative Pre-trained Transformer) [Rad+18], BERT (Bidirectional Encoder Representations from Transformers) [Dev+19], and XLNet [Yan+20] to dominate tasks in Natural Language Generation (NLG). These models have reshaped text generation tasks such as summarization, achieving important results by leveraging attention mechanisms to model complex language structures. BERT, known for its deep bidirectionality, focuses on understanding context from both directions in a sentence, outperforming previous models that considered only unidirectional contexts. XLNet further builds on Transformer-XL by capturing bidirectional context through a permutation-based training objective. In this way, it overcomes some limitations of masked LMs, such as BERT, which assume independence between masked tokens.

The Transformers architecture has three main types: Encoder-Only models (Auto-Encoding Transformers), Decoder-Only models (Auto-Regressive Transformers), and Encoder-Decoder models [RA23]. Each tyoe is suited for different tasks in understanding and generating language. Auto-Encoding Transformers (AET), such as BERT, focus on understanding text. They use the encoder part of the Transformer to look at all parts of a sentence at once, helping them capture meaning from both directions in the text. AET models are trained by hiding some words in a sentence and trying to predict them, which helps in tasks, such as sentence classification and finding named entities (people, places, etc.) in text. Auto-Regressive Transformers (ART), like GPT, use the decoder part of the Transformer to generate text. They predict the next word based on the previous ones, making them great for tasks like text completion and story generation. Finally, Encoder-Decoder models (S2S) use both the encoder and decoder parts of the Transformer. These models are very flexible and can handle tasks that involve both understanding and creating text, such as translation, summarization, and answering questions. Models such as T5 and BART are examples of this, and they work well in different domains, such as healthcare, finance, and social networks.

In addition to being categorized by architecture, Transformer models are also grouped based on their applications. Language-based applications (LBM) focus on tasks such as generating text, summarizing information, and answering questions. Models including

BERT and GPT are fine-tuned for tasks such as sentiment analysis, text classification, and topic modeling, making them useful in many language-related tasks. Domain-based applications (DBM) apply Transformer models to specific fields. For example, BioBERT and ClinicalBERT are used in healthcare for medical document analysis, while FinBERT is used in finance for sentiment analysis and forecasting. In cybersecurity, Transformers help detect malware and other threats. Task-based applications (TBM) focus on specific tasks such as machine translation, named entity recognition (NER), and document summarization. Models including BART and T5 are particularly good at summarizing texts and translating between languages, while DistilBERT, for example, is fine-tuned for answering questions.

Recent advances in medical image captioning have similarly adopted Transformer models. Park et al. [Par+21b] introduced the mDiNAP-transformer and mDiAP-transformer models, both used to create captions for medical images. They both rely on ResNet-152 to extract visual features from normal and patient images. The main difference between them is how they process these features. The mDiNAP-transformer (multi-difference non-average pooling) keeps more detailed information by skipping global average pooling, using features from three layers of ResNet-152 without reducing their size. On the other hand, the mDiAP-transformer (multi-difference average pooling) applies global average pooling, simplifying features from four layers into smaller, one-dimensional vectors. Both models use these processed features in a Transformer decoder to generate reports, with mDiNAP capturing more detail and mDiAP trading some detail for simpler and faster processing. Furthermore, Chen et al. [Che+22] proposed one key Transformer-based model that includes a custom Transformer with a memory-driven unit and generates radiology reports by identifying critical regions of interest in the X-ray images. Another example was used by Xiong et al. [XDY19], which involves a Transformer that utilizes bottom-up attention to detect relevant areas in an image and top-down attention to extract visual features, which are then processed by the Transformer decoder to generate captions.

Attention Mechanisms

Attention mechanisms have become a powerful alternative to traditional Recurrent and Convolutional Neural Networks in tasks such as machine translation, object detection, and image captioning. These mechanisms enable models to focus on specific parts of an input sequence when generating output, regardless of the distance between elements, as introduced by Vaswani et al. [Vas17]. In image captioning, as explained by Xu et al. [Xu+16] and Ghandi et al. [GPM23], attention allows models to focus dynamically on the most relevant parts of an image rather than compress it into a single static vector representation. This mechanism mimics the human cognitive process of selectively focusing on important features in an image, while ignoring less relevant details. By eliminating the need for sequential processing, attention mechanisms capture dependencies across sequences, which can be difficult for recurrent models like RNNs or LSTMs, where information is propagated sequentially through hidden states. According to Ghandi et al. [GPM23], the

attention mechanism computes a context vector that guides the decoder at each time step. This vector is a weighted sum of the encoder's hidden states, where the weights represent the degree of attention given to different regions. The attention vector c_i is defined as:

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{2.1}$$

where h_j is the hidden state of the encoder at time step j, and a_{ij} is the attention weight that indicates how much focus the decoder should place on input$j$ when generating the output at time i.

In encoder-decoder models, as Wu et al. [WC16] highlight, the attention mechanism allows the decoder to access the encoder's hidden states directly, rather than relying solely on the context vector. Therefore, the decoder can focus dynamically on different parts of the input at each step. However, because the decoder typically generates one token at a time, the attention mechanism at each step does not consider future tokens. As a result, when generating complex output, such as captions for images containing multiple objects, the model may initially focus on only one or two objects, missing the overall context. This lack of global context can lead to suboptimal results, especially when tasks require considering multiple elements simultaneously, such as complex images, or involve long-range dependencies, for instance, the dependency between the words "place" and "was" in the sentence: "The place where they had previously met was.."

A popular application of attention mechanisms proposed by Anderson et al. [And+18] in image captioning is the bottom-up and top-down attention model, which enhanced attention mechanisms by incorporating object detection. In this model, attention works on two levels: the bottom-up part detects objects in an image and extracts their features using Faster-RCNN, while the top-down part uses an LSTM-based attention mechanism to focus on the most relevant regions for captioning. At each time step, the model generates a word based on the attended image features and the previously generated words. This dual-layer approach allowed models to pay attention to specific objects and regions within the image and understand their relationships, resulting in more detailed and contextually appropriate captions.

Zohourianshahzadi et al. [ZK22] also presented semantic, spatial, and channel-wise attention, which are techniques that enhance how models interpret images. Semantic attention guides models to focus on specific objects or key features within an image, improving their understanding of what is present. Spatial attention directs the model to important regions of the image, helping it to concentrate on key areas. Channel-wise attention, on the other hand, prioritizes important feature maps, such as color, texture, or

shape, allowing the model to emphasize the most relevant details. By combining these approaches, the models gain a better understanding of both "what" is in the image and "where" to look, leading to more accurate and descriptive output.

Several other models have adapted the Transformer architecture for image captioning. For example, the Meshed-Memory Transformer [Cor+20] introduced memory slots into the attention mechanism, allowing the model to store and retrieve richer feature representations. This approach, combined with bottom-up attention, allowed the model to achieve state-of-the-art results. Additionally, the X-Linear Attention Network [Pan+20] extended this approach by integrating bilinear pooling into the multi-head attention mechanism, allowing the model to capture both spatial and channel-wise interactions in the input features. The multi-head attention mechanism further enhances the model's capability by performing multiple attention operations in parallel. This allows the model to attend to different parts of the sequence with different "heads", capturing various aspects of the input.

2.2 Guided Decoding

Decoding refers to selecting the next word in a sequence based on a probability distribution provided by the model. According to Zarriess et al. [ZVS21], common decoding strategies include greedy search, beam search and nucleus sampling. Greedy search selects the word with the highest probability at each step of the algorithm. While computationally efficient, it often leads to inadequate outputs due to its inability to take the entire sequence of words into account. This constraint can result in incoherent or incomplete texts. On the other hand, beam search is an improved version of greedy search which keeps multiple candidate sequences -called beams- at each step of the beam search algorithm and selects the one with the highest cumulative probability. However, it may still generate repetitive or generic text, especially for longer sequences, as it does not fully address sequence diversity. Nucleus sampling introduces diversity by selecting tokens from the top-P (a specified percentile) of the probability distribution, instead of just picking the most probable token. This method dynamically adjusts the size of the pool of candidate tokens based on how the probabilities are distributed. When the model is confident, the pool shrinks to include only highly probable tokens, whereas for more spread out distribution of probabilities, the pool expands, allowing for more variety. This balances fluency and creativity, avoiding unlikely tokens that could lower the quality of the text.

Although these traditional methods are useful, they often lack control over specific elements of the generated text, such as style, length, or topic relevance. Applying guided decoding can help integrate specific requirements into the text generation process. [Zha+23]. One category of the imposed constraints are the lexical ones, which ensure that specific key-

words or phrases are included in the generated caption. Structural ones are used to control the overall structure of the generated text, such as maintaining sentence length or the use of particular syntactic forms. Another common type is semantic constraints, so that the caption represents a specific theme or emotion, reflecting a positive tone when describing a product. Regarding the paper of Poesia et al. [Poe+22], an example of text generation with semantic constraints, Constrained Semantic Decoding (CSD), is presented. CSD ensures that LMs generate syntactically and semantically correct programs by restricting the next possible token choices during generation. For instance, when generating an SQL query, if a model is required to join the "Flights" and "Airports" tables, CSD constrains the model to select only valid column names (like "AirportCode" from "Flights") to prevent common errors such as mismatching columns between tables. This method ensures that every token generated fits within the defined semantic rules, avoiding issues like undeclared variables or incorrect function arguments.

2.2.1 Guided Decoding for Image Captioning

As discussed in Section 2.2.1, guided decoding is a technique in NLG that shapes the way text is created, by integrating specific constraints, either pre-set or learned during the decoding process [Kal+24]. This way, it differentiates from traditional methods such as greedy search or beam search, which rely only on the model's log-likelihood estimates to select words. In many cases, guided decoding operates during the decoding stage, leaving the pre-trained LM intact while guiding the output towards the constraints [Zha+23].

One particularly important application of guided decoding is in image captioning, a task that involves generating a descriptive text based on an image. The goal is to produce fluent, coherent sentences that accurately reflect the content of the image and also respect specific constraints. Traditionally, this is done using sequence-to-sequence models with attention mechanisms that decide which part of the image to focus during caption generation. However, many real-world applications require greater control over the attributes of the captions produced, making the integration of Controllable Text Generation (CTG) approaches vital [Zha+23]. The Input-Process-Output (IPO) framework, illustrated in Figure 2.3, provides a clear structure to understand how controlled elements can guide text generation to meet specific requirements.

2.2.2 Methods to implement Guided Decoding in Text Generation

There are different techniques to guide and control the text generation process towards desired outputs. Many of them use weighted decoding, a specific type of guided decoding. In practice, the output of the LM is controlled by adjusting token probabilities based on

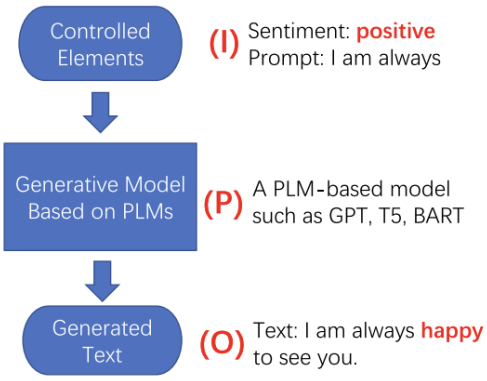

Fig. 2.3: This figure shows the Input-Process-Output (IPO) structure of a controlled text generation system. The input (I) consists of controlled elements, such as a condition (e.g., sentiment: positive) and a source text prompt. The process (P) uses a Pre-trained Language Model (PLM) like GPT, T5, or BART to generate text. The output (O) is the final text that satisfies the input condition, such as "I am always happy to see you" for a positive sentiment. This figure was created by Zhang et al. [Zha+23].

specific rules or constraints, without modifying the original model. Instead, the output is guided in a post-hoc manner [Zha+23]. For instance, if a generated sentence must include a specific keyword or a particular sentiment, the probabilities of relevant tokens are increased, while the probabilities of less relevant tokens are reduced. On the other hand, many techniques focus on optimizing the generation process through Reinforcement Learning (RL), which involves learning from feedback signals. RL-based methods are key methods for changing the original model, as they adjust the model's internal parameters during training. They treat text generation as a sequential decision-making process, where a reward function guides the generation towards a desired output. By optimizing this reward function, which may represent user preferences or relevance, the model learns to dynamically adjust its output [KLM96].

One such method is Discriminator-Guided decoding [Zha+23]. It is an approach in which a separate classifier (discriminator) is trained to evaluate whether the generated caption matches certain requirements, e.g. including specific words or having a positive sentiment. The discriminator continues to give feedback during the decoding process, thus adjusting the probabilities of the next token to favor those that align with the imposed constraints. This approach ensures that the caption adheres to both the visual content of the image and the desired control constraints. Generative Discriminators is an example of Discriminator-Guided decoding. It uses generative discriminators like GeDi [Kra+20], which use small, task-specific models trained to distinguish between controlled and uncontrolled text. In the context of image captioning, a generative discriminator could help prioritize captioning candidates that match certain criteria or contain certain key elements from the image.

Rashid et al. [Ras+24] also explore another decoding method, inspired by RL, called reward-guided text generation (RGTG). This method leverages a reward model to score partial

sequences during the generation process, steering text generation toward higher-reward sequences. Also inspired by RL, Jiang et al. [Jia+18] presented a method using the guiding network, in the context of image captioning. This network provides a guiding vector, which is used at each step of the decoding process to make the captions more precise and relevant. This vector integrates image attributes with learned language patterns and allows for adjustments to sentiment, style, or even specific keywords, thus being useful whenever captions must meet specific lexical, structural, or semantic requirements [Zha+23]. For example, in a marketing scenario, a generated caption may need to emphasize positivity or excitement about a product depicted in the image. Without this additional guidance, standard decoder models could possibly produce generic or overly simplistic captions that may not capture the most prominent elements of the image.

In terms of weighted decoding, Zhang et al. [Zha+23] describe manipulating the decoding process to meet specific goals without altering the original LM, as demonstrated by Plug-and-Play models such as PPLM (Plug-and-Play Language Models) [Dat+20]. They work by allowing an external system to re-rank the generated tokens based on the desired control attributes and then adjust the captions generated by a pre-trained model. The image captioning model generates candidate captions and the plug-and-play model adjusts these candidates by modifying the LM's hidden states to prioritize tokens that align with specific requirements (such as toxicity reduction).

In addition, heuristic-based approaches, which can be considered as a form of weighted decoding, can also serve the purpose of generating text without altering the original model. These methods rely on human-made rules or weighted features to influence the word choice at each step of generation. An example of guiding the decoding process was presented by Wu et al. [Wu+16] when they used a heuristic function for length normalization that improves translation quality in Google's Neural Machine Translation (GNMT) system. This length normalization heuristic adjusts the log-probability scores of translations and is used to guide the beam search decoding process and prevent the search from being biased toward shorter translations. In addition, they introduced a coverage penalty, which encourages the model to fully attend to all parts of the source sentence, penalizing translations that ignore or overemphasize certain parts.

Another example of weighted decoding is the FUDGE method [YK21]. FUDGE improves the efficiency of the generation process by using a Bayesian decomposition to predict desired features, without needing multiple iterations of adjustments. This method works by training a lightweight future discriminator that estimates whether a partial sequence will lead to the desired attribute in the future. FUDGE then adjusts token probabilities dynamically in real-time, focusing on the log-probability space for numerical stability, and can handle multiple independent attributes by summing their respective probabilities. In addition, it achieves controlled text generation without modifying the underlying LM, making it adaptable to various models and tasks.

2.3 Co-occurrence of Terms for Caption Retrieval as a Foundation for Guided Decoding

The field of Information Retrieval is evolving to develop new methods that improve the relevance of the retrieved information. Traditional term-based retrieval systems rely on the exact match among the query terms and the terms found in documents or sentences. As defined by Sneiders et al. [Sne24], another technique is based on vector-space models, where documents and queries are represented as vectors in a high-dimensional space. These models, including the widely recognized cosine similarity model by Salton and McGill (1986) [Sal83] and the probabilistic retrieval model by Robertson and Spark Jones (1976) [RJ76], have been at the core of information retrieval for decades. However, these models typically emphasize the presence of individual terms (using TF-IDF), often overlooking the significance of them coexisting within the same sentence, which results in a failure to provide the specific context needed.

Recent research has increasingly focused on enhancing the relevance scoring in text retrieval by incorporating factors beyond just individual term frequency and document frequency. Either in document or in sentence retrieval, the co-occurrence of terms has been already used as a technique to filter the information retrieved. Term co-occurrence considers pairs of words that appear together within a specific context (such as a sentence or a paragraph). In terms of sentence, this approach leads to the retrieval of sentences, which contain not only one individual term, but usually two or more. The outcome of this procedure is the capture of more meaningful and contextually relevant material than when considering isolated terms. For example, the coexistence of terms “summer” and “job”, indicates a specific context related to employment opportunities during summer. Ensuring that both words appear in the same sentence helps the algorithm render the intent and meaning behind a user’s query and to distinguish between different contexts in which only one term might be used.

The work of Sneiders et al. [Sne24] explores the effectiveness of term co-occurrence, particularly for the task of matching email-style query messages to webpages, without relying on extensive machine learning models or predefined knowledge bases. The approach involves splitting documents into smaller chunks, like sentences or paragraphs, and focusing on pairs of terms that appear together in both the query and the document. This helps filter out irrelevant documents and narrows the search to a smaller set of highly relevant ones. This approach contrasts with traditional methods that treat documents as bags-of-words, disregarding the structural information within the text. Experiments conducted showed that using term co-occurrence significantly improved the recall and precision of relevant document retrieval compared to traditional bag-of-words models.

Also, Mittendorf et al. [MMS00] used co-occurrences of words within predefined text windows (such as sentences or paragraphs) as indexing features for document retrieval, rather than relying exclusively on individual word-based features. This method is based on the idea that the closer query terms appear together in a document, the more likely that document is to be relevant to the query. In order to test this, the commonly used Robertson-Sparck Jones (RSJ) weighting was applied to both first-order features (individual words) and second-order features (co-occurrences of words within predefined text windows) within the probabilistic retrieval framework. The experiments were carried out to test their ability to rank relevant documents effectively. The results revealed the effectiveness of co-occurrences within sentence-size and paragraph-size windows, which significantly outperformed first-order, word-based features, particularly for larger query sizes (greater than 10 features). This indicates that local co-occurrences of terms are stronger indicators of relevance than the mere presence of individual words.

According to Ferilli et al. [Fer+10], one methodology which leverages co-occurrence for retrieval is to build a modified vector space, where related terms are given significance even if they do not appear directly in the document. The process involves constructing a term-term matrix that records how often terms co-occur across the document collection. This matrix is then combined with the original term-document matrix, adjusting the weight of terms in each document according to not only their presence but also their connection to other relevant terms in the query. This approach has proven effective in retrieving documents where the exact query terms are missing, but related terms are present.

Ferilli et al. proposed two specific techniques for implementing this co-occurrence-based retrieval: SuMMa (Successive Multiplication of Matrices) [VW97] and LSE (Latent Semantic with Explicit co-occurrences) [Fer+10]. The SuMMa approach introduces co-occurrences into the vector space by using matrix operations. First, a term-term matrix is constructed to capture the co-occurrences of terms across the document collection. This matrix is then used to modify the original term-document matrix, re-weighting the importance of terms based on their co-occurrence with other related terms. The resulting modified matrix allows for the retrieval of documents that may not contain the exact query terms but other relevant ones. The LSE approach is inspired by Latent Semantic Indexing (LSI) [KP06] and focuses on the explicit incorporation of term co-occurrence in the vector space. The technique involves performing Singular Value Decomposition (SVD) on the term-document matrix and then reconstructing the matrix with an emphasis on the most significant latent concepts. By emphasizing the co-occurrence of terms in this process, the LSE method enhances the retrieval model's ability to capture the underlying semantic relationships between terms, improving search accuracy.

3 System Design and Implementation

3.1 Baseline

The methods implemented in this thesis build upon the work by Kaliosis et al. [Kal+23], which focuses on generating diagnoses for biomedical images. These images are associated with specific biomedical tags, which carry important information that should be depicted in the final diagnosis. The whole approach of Kaliosis et al. [Kal+23] is based on the idea that the quality of a diagnostic caption is strongly associated with how effectively the generated report incorporates the image's key medical tags. Following this, the goal of the implemented method is to alter the beam score at each step of the beam search algorithm. In this way, it will guide the step-by-step generation of diagnostic captions toward expressing the tags in the generated caption as explicitly or implicitly they are expressed in the training captions. This process notably falls under the scope of Guided Decoding, where the algorithm is guided by training data.

The training data consist of biomedical tags, each of which is associated with a specific set of training captions. Before executing the beam search algorithm, they compute for each tag and its associated captions the Maximum Cosine Similarity (MCS) scores. MCS(t, c) between a tag t and a caption c is computed as follows:

$$\text{MCS(t, c)} = \max_{1 \leq j \leq |c|} sim(h(t), h(w_j^c)) \tag{3.1}$$

In Equation 3.1, $sim(h(t), h(w_j^c))$ measures the cosine similarity between the embedding of the tag t and the embedding of the j-th word w_j^c in the caption c, with $h(\cdot)$ representing the embedding function.

Given a tag and a set of tokens that compose the caption c, the algorithm calculates the MCS score between the tag and each individual token in the caption. The highest similarity score among these values demonstrates how explicitly the tag is expressed in at least one token of the caption, assuming that a higher cosine similarity suggests a stronger semantic correlation between the tag and the word. Figure 3.1 displays a heatmap where the words of the caption are represented on the horizontal axis, while the associated tags are shown on the vertical axis. Each block of the heatmap represents the cosine similarity between

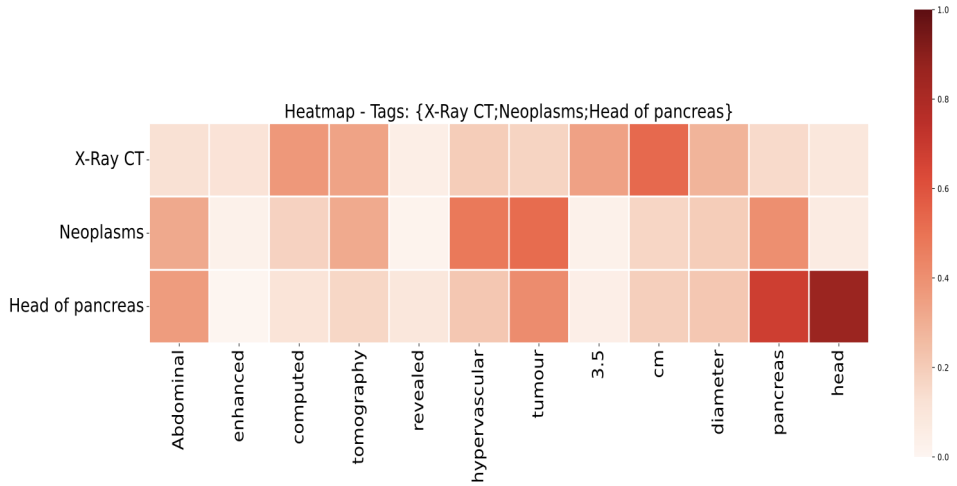

Fig. 3.1: Heatmap that visualizes the cosine similarities between the words of a ground truth caption (x-axis) and its associated biomedical concepts (y-axis), created by Kaliosis et al. [Kal+24].

the word embeddings of the tag and the word token embeddings in the caption, with larger values represented by darker blocks. For instance, when the tag is "Head of pancreas" and the token in the caption is "head", as shown in Figure 3.1, their cosine similarity will be high, indicating that the tag is almost explicitly expressed in the caption.

Computing the MCS score between a tag and every associated training caption leads to a distribution of MCS scores for each tag. Taking the median value of this distribution results in the Median Maximum Cosine Similarity (MMCS) score, which reflects how explicitly the tag t is expressed on average among all its associated training captions S. The MMCS is computed using the following formula:

$$\text{MMCS(t)} = \text{median}\left(\{MCS(t, c) \mid c \in C\}\right) \tag{3.2}$$

The MMCS score determines the explicitness level of each tag, empirically derived from the captions associated with that tag. Tags with a higher MMCS score are almost explicitly mentioned throughout their associated training captions, while those with lower scores are likely expressed in a less explicit manner. Once these scores are computed for all available tags in the training set, an alternation of the beam search algorithm that uses these scores is executed. This algorithm step-by-step constructs the generated caption of the given image.

At each step of the algorithm, given an image and a set of medical tags[1] that describe it, the medical image captioning system evaluates how explicitly each tag t is mentioned in the already generated caption c, by computing the MCS score, as described earlier. For example, when the image includes the tag "Heart" and the current generated text includes

[1]These tags could be ground truth or system-generated predictions

the word “Heart”, the algorithm returns an MCS of 1, due to the direct match. In contrast, if the text uses a related but less explicit term such as “cardiac” the algorithm returns a lower MCS score, reflecting the more implicit nature of the reference.

By minimizing the difference between the MCS and MMCS scores, it is ensured that the generated caption expresses the tags with a similar degree of explicitness as seen in the training data. To achieve this, MCS and MMCS scores are used to impose a penalty at each decoding step, forcing the decoder to choose among the captions generated by the search algorithm that encapsulate the associated biomedical tags more or less explicitly, depending on the training captions. Given a set of tags T and a caption c, the imposed penalty is calculated as:

$$\text{DMMCS}_\rho(T, c) = \frac{1}{|T|} \cdot \sum_{t \in T} (MCS(t, c) - MMCS(t))^2 \tag{3.3}$$

At each decoding step, every candidate caption c generated by the decoder is scored and pruned using a modified version of the standard beam search algorithm based on the following formula:

$$\text{DMMCS}(c) = \alpha \cdot \text{DMMCS}_\rho(T, c) + (1 - \alpha) \cdot (1 - D_{\text{score}}) \tag{3.4}$$

Here, D_{score} represents the beam score computed by the standard beam search algorithm, typically calculated as:

$$D_{\text{score}} = \frac{1}{|c|^\beta} \sum_{i=1}^{|c|} \log p(c_i | c_{<i}) \tag{3.5}$$

where $p(c_i|c_{<i})$ represents the conditional probability of the i-th word c_i given the previously generated words $c_{<i}$, and β is a length normalization parameter. The weighting factor α balances the impact of the score computed by the proposed DMM algorithm and the standard beam search score. A higher value enforces the model to give more importance to the DMM component and the tag expression is aligned with the training data used by the algorithm, while a lower value favors the probability-based choice of the standard beam search algorithm. This allows for fine-tuning based on the desired level of explicitness in the generated captions.

3.2 Tuples-DMM: A data-driven decoding method using tuples of tags

This section describes a method that aims to alter the penalty imposed by the baseline DMMCS method at each decoding step during the generation of diagnostic captions. The main idea is to filter the training captions used to guide the decoding algorithm, which

are responsible for the computation of the penalty. Practically, the Tuples-DMM method works as an **information retrieval technique**, which forms pairs of two tags by the available tags associated with an image and retrieves from the training set, for each pair, the **common associated captions** between its two tags. It leverages only these specific captions to guide the algorithm, as they are considered contextually significant. The method falls under the broader category of Information Retrieval and NLP, with a specific focus on contextual relevance between tags and captions, as well as on the exploration and retrieval of captions where two medical tags co-occur.

3.2.1 Tuples-DMM method explanation

As mentioned previously, each image to be captioned is associated with tags likely to be expressed in the generated caption either explicitly or implicitly through semantically related words. The baseline method focuses on expressing these individual tags in the generated caption at the same level of explicitness as they are expressed in their associated training captions. For example, if the training captions associated with the tag "Heart" frequently use a contextually related term such as "Pericardial", the algorithm will aim to reflect this level of explicitness in the generated caption. However, training captions where only a single tag exists –without considering the co-occurrence with other tags– may constitute captions that are contextually generic or irrelevant to the specific image context. A retrieved caption can be considered generic or irrelevant when it does not clearly represent the specific medical condition of the image or fails to use the necessary medical terminology. For instance, the medical term "Heart" can be associated with captions describing various conditions, not related to each other. Therefore, retrieving all the associated captions may guide the algorithm towards irrelevant concepts. This is the reason why the proposed method does not treat tags as individual entities but instead considers them in tuples.

This approach leverages the **co-occurrence of tags** to retrieve richer information than when considered individually. Specifically, it considers tuples of two tags and examines only their **common associated captions** to compute the imposed penalty. The pairs are formed from all possible combinations of available tags associated with an image. For example, if the set of available tags is:

$$\{\text{"Heart"}, \text{"X-Ray"}, \text{"Liver"}\},$$

the created tuples will be the combinations:

1. ("Heart", "X-Ray")
2. ("Heart", "Liver")
3. ("Liver", "X-Ray")

Captions containing both "Heart" and "Liver" are expected to provide more relevant and contextually meaningful information compared to captions containing only "Heart" or only "Liver". This is because captions mentioning both tags are likely to describe medical conditions or findings that involve both organs, whereas captions with just "Heart" or "Liver" may refer to unrelated conditions that do not connect the two organs.

In general, by narrowing the retrieval to the intersection of text captions that contain both tags, the sample space becomes significantly smaller, ensuring that the items within this space are contextually relevant and closely aligned with the combined medical context represented by the tag pair. Consequently, the beam score can be adjusted with greater precision and the generation process is steered toward constructing captions that are not only valid, but also contextually accurate, as they are more likely to describe the specific medical concepts that both tags are involved. In the biomedical domain, the reference of specific terminology and the highlighting of medical conditions is a crucial part of a patient's diagnosis generation. At the same time, referencing an unrelated condition or context that is not pertinent to the intended meaning is very risky. By focusing on tag pairs and their common captions, the method ensures that the retrieved information is more likely to be accurate and relevant to the specific medical condition that we need to showcase, reducing the chances of including misleading content.

3.2.2 Computation of Tuples-DMM method

As already mentioned, the method aims to modify the computation of DMMCS penalty imposed during the modified version of the beam search algorithm. The penalty value increases as the difference between how tags are expressed in the training captions and the generated caption grows, with a higher penalty indicating greater inconsistency in tag expression.

Leveraging the Training Data

This subsection describes how the training images, each associated with multiple tags and a caption, are processed. The goal is to extract certain measures that will later be used during inference, when generating a caption for a new image.

For every training image, all possible pairs of two tags are created. Then, iterating over each tuple, we calculate a distinct metric, the **Median Maximum Cosine Similarity (MMCS)** score, for each tag in the pair, which indicates how explicitly the tag is represented in their common associated training captions C. To compute the MMCS score, a series of **Maximum Cosine Similarity (MCS)** scores is first calculated between each tag in a tuple and each caption c_j in C. An MCS score is calculated as the cosine similarities between a tag and each token within a caption, then selecting the maximum value that

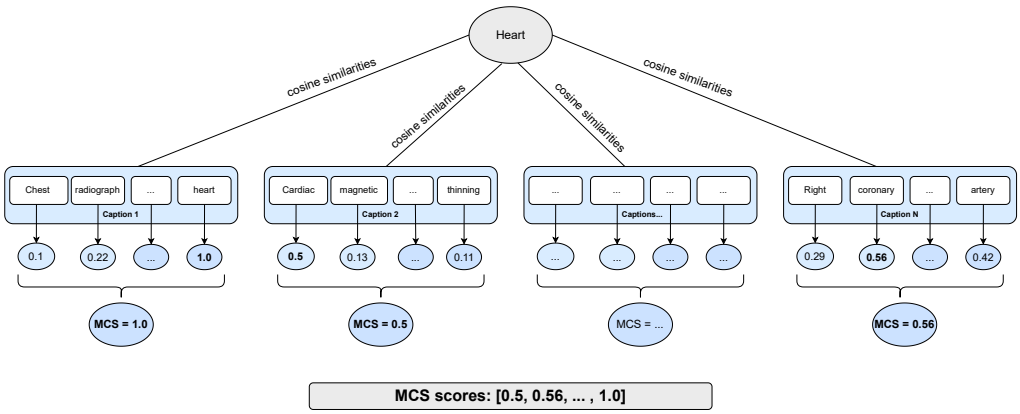

Fig. 3.2: This example demonstrates how the distribution of MCS scores is computed for a specific tag. Taking the tag "Heart" as an example, the cosine similarity is calculated between the tag and each token within an associated caption. The highest cosine similarity value among the tokens is selected as the MCS score for that caption. Repeating this process for all captions generates a distribution of MCS scores.

represents the highest correlation between the tag and any token in the caption. In Figure 3.2, we can see the tag "Heart" and the selection of the maximum cosine similarity score between the tag and each token within the associated captions (e.g. Caption 1 returns an MCS score of 1.0). The MCS score for a tag is defined as:

$$\text{MCS}(\text{tag}, T) = \max_{i=1,\ldots,n} (\text{cossim}(e(\text{tag}), e(t_i))) \tag{3.6}$$

where T is the set of tokens in the caption, n is the number of tokens in T, t_i is the i-th token of the caption, cossim() denotes the cosine similarity function, and $e(\text{tag})$ represents the embedding of the tag.

A high MCS score suggests that at least one token in the caption is contextually strongly correlated with the tag, and if this similarity score is close to or equal to 1, it indicates that the tag is explicitly mentioned in the caption. The difference in the proposed method compared to the baseline is that the computation of MCS scores is guided only by the common associated captions of the tags that form a tuple. Calculating, for each tag in the tuple, one MCS score per caption c_j leads to a **distribution of MCS scores** for each tag, revealing how it is represented across all common captions c_j in C. Consequently, for each tuple , we obtain two distributions of MCS scores—one for each tag. To illustrate it using an example, we assume that there is a tuple that includes "Heart" (e.g., ("Heart", "Blood Vessel")). As shown in Figure 3.2, calculating the MCS scores of the tag "Heart" for each common associated caption (Caption 1, Caption 2, ... , Caption n), produces a distribution of MCS scores for the tag "Heart" ([0.5, 0.56, ... , 1.0]). Another distribution is also calculated for "Blood Vessel".

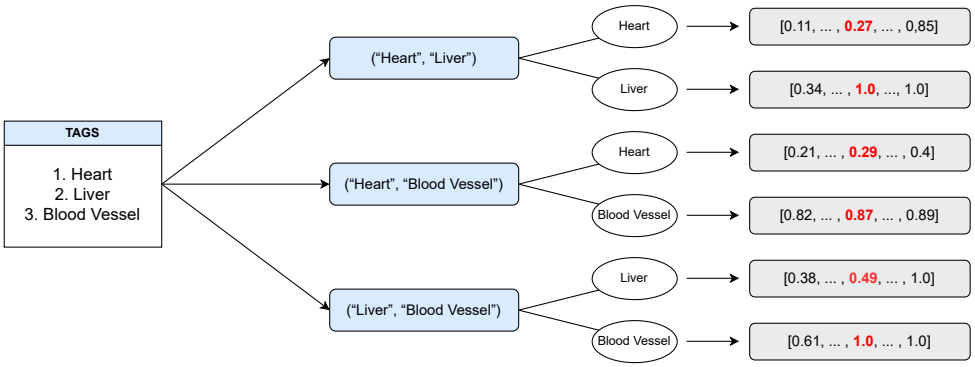

Fig. 3.3: This diagram shows how MMCS scores are calculated for tag pairs. From three tags, three tuples are created: ("Heart", "Liver"), ("Heart", "Blood Vessel"), and ("Liver", "Blood Vessel"). For each tag in a tuple, the MCS distribution is computed, and the median value (red circles) of the distribution is taken as the MMCS score, providing a measure of how the tag is represented in the captions associated with the specific tuple.

The **MMCS** score is the **median** value of the distibution of MCS scores for each tag of a tuple and represents how explicitly the tag is expressed in these filtered captions, retrieved for the specific tuple. For each tag of the tuple, the score is computed during the training process this way:

$$\text{MMCS}(\text{tag}, C) = \text{median}_{j=0}^{n} \left(\text{MCS}(\text{tag}, c_j)\right) \tag{3.7}$$

where C is the set of common associated captions between the two tags, n is the number of captions in C, and c_j is the j-th caption in C.

As shown in Figure 3.3, for each tuple created by the available tags, the MMCS values are calculated and stored separately for each tag (e.g., for the tuple ("Heart", "Blood Vessel"), one value is stored for "Heart" and one for "Blood Vessel"). These scores are then applied during the modified beam search algorithm to guide generation. Notably, if a tag appears in multiple tuples, such as "Heart" in both ("Heart", "Blood Vessel") and ("Heart", "Liver"), it may receive different MMCS scores for each tuple (e.g., 0.62 in the first tuple and 0.86 in the second). This variation arises because each score is derived from the specific captions associated with each different tuple, to which the tag belongs.

Imposing the penalty during caption generation

Before executing the modified version of the beam search algorithm, we have computed the **MMCS** score using the **training captions**. When generating a caption for a new image using the search algorithm, we first form all the possible tuples from the associated tags. Then, we assess the semantic correlation between the tuple and the partially generated caption at each step. To do this, for each tag in the tuple, we calculate the **MCS** score

between itself and the **currently generated caption**. Intuitively, the MCS value, updated dynamically during the beam search process, indicates how explicitly the tag is represented in the newly generated caption.

Using the MMCS score from the training set and the MCS score calculated during the execution of the beam search algorithm, we compute their difference for both tags of each tuple:

$$\Delta_{\text{MCS}}(\text{tag}, \text{caption}) = \text{MCS}(\text{tag}, \text{caption}) - \text{MMCS}(\text{tag}) \tag{3.8}$$

When this difference Δ_{MCS} is equal to zero, the tag is expressed in the generated medical caption exactly the same way as it is expressed in the training captions. This is the goal that the TDMM penalty is designed for; rewarding cases where Δ_{MCS} tends to zero. Given a set of tuples T and a generated caption c, the penalty imposed is defined as:

$$\text{TDMM}_{\rho}(|T|, \text{caption}) = \frac{1}{|T| \times 2} \sum_{i=1}^{|T|} \sum_{j=1}^{2} \left(\Delta_{\text{MCS}}(\text{tag}_{i,j}, \text{caption})\right)^2 \tag{3.9}$$

where,
$|T|$ is the number of tuples,
$\text{tag}_{i,j}$ represents the j-th tag in the i-th tuple in T,
and caption is the generated caption.

The penalty formula sums the squared differences for each tag across all tuples, then averages by dividing by $|T| \times 2$, where $|T|$ is the number of tuples, and each tuple contains two tags. This averaging makes the penalty independent of the number of tuples. By squaring each difference, the formula gives more weight to larger mismatches, penalizing the cases where the generated caption does not match the tag expression of the training data. In this way, the TDMM penalty is minimized when the generated caption aligns closely with the training captions, so it is more likely that the particular beam will be selected by the algorithm. To further illustrate the imposition of this penalty, Figure 3.3 presents an example with the tuples (“Heart”, “Blood Vessel”), (“Heart”, “Liver”), and (“Blood Vessel”,“Liver”). In this example, the calculated penalty sums Δ_{MCS} are six, for each tag appearance in these tuples. Thus, MCS − MMCS is computed twice for every tag as each tag is part of two tuples.

One tag (e.g “Heart”) will have different MMCS scores, depending on the tuple it belongs to. The variation of the MMCS score for the same tag rewards the cases where a tag is similarly represented in the associated captions of at least one of the tuples. For example, in Table (A) of Figure 3.4, the tag “Heart” is explicitly mentioned in the retrieved captions for both tuples containing it, so the MMCS for both tuples equals 1.0. In this case, if the generated caption also includes “Heart”, the MCS would reach 1.0, and the difference between MCS and MMCS would be equal to zero. On the other hand, as shown in Table (B)

(A)

Tuple	Tags	
(Heart, Liver)	**Heart = 1.0**	Liver = 0.72
(Heart, Blood Vessel)	**Heart = 0.5**	Blood Vessel = 0.33

(B)

Tuple	Tags	
(Heart, Liver)	**Heart = 1.0**	Liver = 0.72
(Heart, Blood Vessel)	**Heart = 1.0**	Blood Vessel = 0.33

Fig. 3.4: Each table presents how the tags of two tuples are represented throughout their common associated captions (MMCS scores). In Table (A), "Heart" is explicitly mentioned in the training captions of both tags. However, in Table (B), "Heart" is explicitly represented in the captions of tuple ("Heart" , "Liver") but implicitly represented in the captions of tuple ("Heart" , "Blood Vessel")

of Figure 3.4, the MMCS score for the training captions is 0.5 for one tuple (tuple: ("Heart", "Blood Vessel")) and 1.0 for the other (tuple: ("Heart", "Liver")). In that case, the algorithm rewards the generated caption if it includes either "Heart" with a similarity score of 1.0, or a term with a similarity score of 0.5, such as "cardiac". Thus, the algorithm rewards both forms of tag expression observed across different set of training captions which may highlight different medical concepts.

Overall, this methodology, focusing on the co-occurrence of two tags in the same caption, aims to retrieve the most contextually relevant training captions that include both tags. It uses only these captions to compute the beam score, filtering out irrelevant misleading captions. For instance, (Figure 3.5), if the tag "X-Ray Computed Tomography" has 24,695 associated captions and the tag "Pelvis" has 3,063, but only 1,107 captions contain both tags, the method focuses exclusively on these 1,107 captions, which are more likely to describe concepts related to Pelvis X-Rays. The desired result of the methodology is to guide the modified beam search algorithm to ensure that the tags are expressed with the appropriate level of explicitness in the final generated caption. This retrieval process reduces the number of captions considered and narrows down the context of the retrieved captions. It is important to note that the standard beam search algorithm is still considered during generation; however, the Tuples-DMM method helps the algorithm focus on specific patterns in the training data, ensuring that key medical terminology and conditions are accurately incorporated into the final diagnosis.

3.2.3 Tuple-Based Retrieval with Dissimilarity Maximization

An important observation regarding the Tuples-DMM methodology is that when the two tags forming a tuple are highly similar contextually, the retrieval process may still include irrelevant captions. This phenomenon can occur because similar tags are likely to be

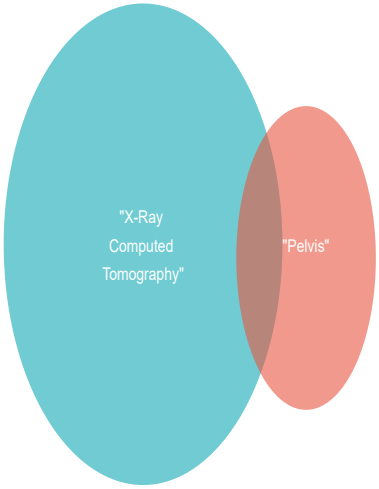

Fig. 3.5: This Venn diagram demonstrates how the common captions associated with two tags are fewer than their individual associations. "X-Ray Computed Tomography" is linked to 24,695 captions, "Pelvis" to 3,063 captions, but only 1,107 captions are associated with both tags. These filtered captions are considered to provide more useful information.

associated with the same captions, which diminishes the effectiveness of the filtering process. As shown in the table in Figure 3.6, the tuple formed by "Chest" and "Plain Chest X-Ray", which are contextually similar, retrieves a high number of associated captions. However, in the case of tags such as "Heart" and "Liver" or "Pleura" and "Liver", the retrieved information is significantly reduced. This reduction helps the algorithm focus on the specific concept that combines these different medical terms. This observation led to an enhanced approach, where not only are tuples of tags used to retrieve common captions, but these tuples are specifically chosen so that their tags are contextually distinct.

This additional approach of Dissimilarity Maximization suggests that when forming tuples, a dissimilarity measure is applied to potential tag pairs. This measure is the cosine dissimilarity of the tags that compose the tuple in the semantic embedding space. Only pairs of tags with a high dissimilarity score are selected as the final tuples used by the algorithm. This technique takes advantage of the idea that information fitting into two very different contexts is likely to be more useful and relevant to the specific medical condition depicted in the image.

3.2.4 Tuple-Based Retrieval with Extraction of Frequent Tags

An additional technique to further enhance the filtering process of the retrieved information is the extraction of frequent tags when creating tuples. By identifying and excluding

Tuple - # associated captions	Tags - # associated captions	
(Chest, Plain Chest X-Ray): 2908	Chest: 8199	Plain Chest X-Ray: 3451
(Pleura, Liver): 2	Pleura: 88	Liver: 1136
(Heart, Liver): 6	Heart: 1096	Liver: 1136

Fig. 3.6: This table presents the count of retrieved captions where two tags coexist, as well as the number of separate captions associated with each tag individually. The first row shows a tuple created from contextually similar tags, in contrast to the next two rows, which display tuples formed from dissimilar tags. This highlights that maximizing dissimilarity in tuple creation significantly reduces the information utilized by the algorithm.

frequent tags, we focus on forming tuples with infrequent tags, which tend to carry more specific and significant information.

This approach is based on the observation that infrequent words often represent more specialized or contextually important information, especially in domains such as medical text. Examples of such tags are shown in Figure 3.8, such as "Ligaments" and "Axilla". These tags are not associated with many captions and represent medical terms, conditions, or procedures that help the algorithm focus on specific medical topics. Common tags, while also useful, often represent broad and general concepts that may not display the specific concept of the image being used for caption generation. Some examples are shown in Figure 3.7, such as "Plain X-Ray", "X-Ray Computed Tomography", "Magnetic Resonance Imaging", which are associated with a large number of captions, as you can, for instance, find MRIs for different parts of the human body. By focusing on tuples formed from these less common tags, this approach mitigates the risk of the generated captions being overly generic or missing critical details.

Tag Name	Count
X-Ray Computed Tomography	24,695
Plain X-Ray	19,833
Magnetic Resonance Imaging	11,554
Ultrasonography	9,949
Chest	8,199
Anterior-Posterior	7,153
Contrast used	5,854
Angiogram	4,707
Bone structure of cranium	3,456
Plain chest X-Ray	3,451
Abdomen	3,368
Postero-Anterior	3,356
Pelvis	3,063
Lower Extremity	2,431
Sagittal	2,278

Fig. 3.7: Top 15 Tags with the Highest Number of Associated Captions

Tag Name	Count
Fibula	100
Right frontal lobe structure	99
Nose	99
Left breast	99
Brain Stem	99
Subcutaneous Tissue	99
Ischemic	98
Upper abdomen structure	98
Ligaments	98
Stomach wall structure	98
Structure of wisdom tooth	98
Axilla	98
Head and neck structure	97
Diffusion weighted imaging	97
Structure of condyle	97

Fig. 3.8: Tags with Number of Associated Captions Below 100

4 Data

Biomedical datasets are fundamental to the advancement of medical image analysis. In the context of this thesis, we used the ImageCLEFmedical 2023 dataset [Rüc+23], which offers a diverse collection of radiology medical images and corresponding diagnoses. The following sections will provide a detailed overview of this dataset to better understand its characteristics before applying the methods discussed in Sec. 3 to the data.

4.1 Dataset Overview

The dataset is divided into three main subsets. The **training set**, consists of 60,918 radiology images with 263,091 concept occurrences and 2,125 unique Unified Medical Language System® (UMLS) [Nat23] concepts. These concepts, otherwise referred to throughout this thesis as "tags", are keywords or phrases associated with specific radiology images, maybe redundant. The **validation set** contains 10,437 images associated with 46,584 concept occurrences and 1,945 unique concepts. Finally, the **test set** consists of 10,473 images with a total of 46,955 concept occurrences and 1,936 unique concepts. Special attention was paid to refining medical image modality concepts (e.g., X-ray, MRI, CT), and for X-ray images, additional anatomical and directionality concepts were included, such as coronal, posteroanterior (PA), anteroposterior (AP), sagittal, or transversal views.

This dataset was published for the ImageCLEFmedical 2023 [Rüc+23] competition, which aims to advance medical image understanding through two primary subtasks: **concept detection** and **caption prediction**. The first task focuses on automatically extracting Unified Medical Language System (UMLS) concept annotations and the second on generating descriptive captions from radiology images.

4.1.1 Concept Detection

The concept detection subtask in ImageCLEFmedical 2023 aims to extract the Unified Medical Language System® (UMLS) Concept Unique Identifiers (CUIs) from radiology images. These CUIs are codes that correspond to biomedical terms —words or phrases— referred to as “concepts” or “tags”. For example, the CUI "C0040405" represents "X-Ray Computed Tomography". Among the extensive set of available concepts, several are directly related to imaging modalities, including X-Ray Computed Tomography, Ultrasonography,

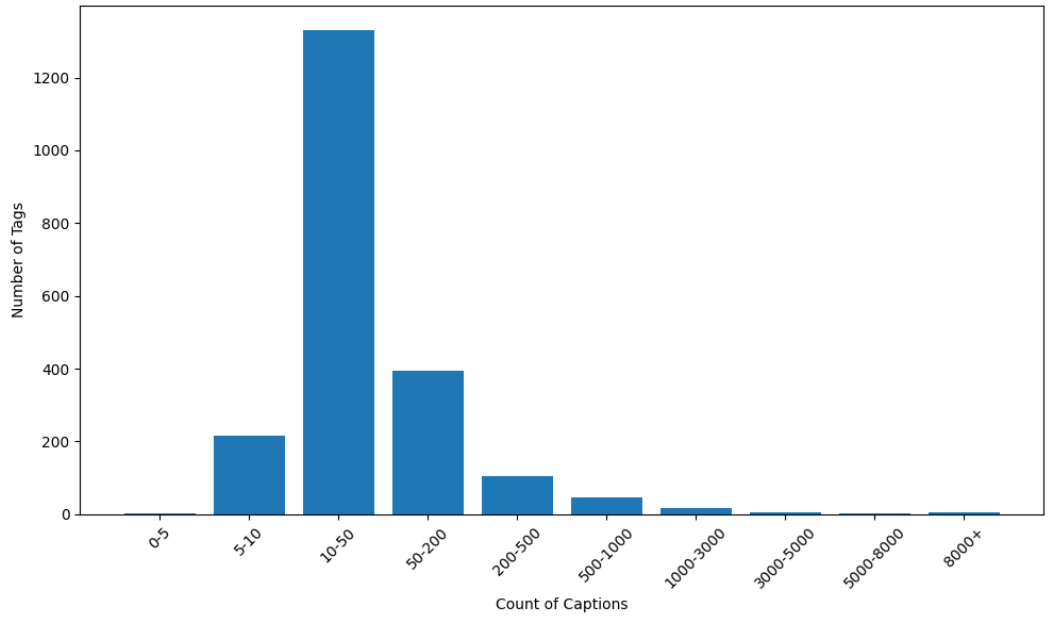

Fig. 4.1: This bar chart has the x-axis representing specified ranges referring to the count of captions, and the y-axis representing the number of tags associated with a corresponding range of captions.

Magnetic Resonance Imaging, and PET/CT scans. Each image may be associated with multiple concepts, meaning an individual image can have several relevant tags reflecting the medical conditions presented in the dataset.

4.1.2 Caption Prediction

The caption prediction subtask aims to generate coherent and descriptive textual captions for radiology images. Each image is linked to a unique ground truth caption, resulting in a dataset of 71,355 captions, of which 99.46% are distinct. The challenge is to produce captions that accurately reflect the content of images using advanced machine learning models. The model performance is evaluated by comparing the generated captions to the ground truth using specific metrics. The captions vary significantly in length, ranging from one word (occurring 134 times) to a maximum of 315 words (encountered once), with an average length of 16.04 words. To promote concise and accurate diagnostic generation, captions longer than 90 words were excluded during training, mitigating potential hallucinations by Large Language Models (LLMs).

For the caption generation task, there is a set of 2,125 concepts, which exhibit significant imbalance. As shown in Figure 4.1, certain tags are associated with thousands of images, some exceeding 8,000 occurrences, whereas the majority are linked to fewer than 50 images. This imbalance inspired the idea of extracting frequent words when generating captions, as they seem to cause the algorithm to focus on a large number of non-informative captions.

On average, each image used during caption prediction is associated with 3.74 tags, with a minimum of one tag and a maximum of 32 tags per image. This subtask addresses the critical need for the appropriate leveraging of tags to generate accurate textual descriptions that can aid in medical image understanding.

5 Evaluation

5.1 Evaluation Metrics

This section presents the results of the methods implemented in this thesis. Using the DMMCS methodology proposed by [**Kal+23**] as a baseline, we conducted experiments to evaluate the effectiveness of various versions of the **Tuples-DMM** approach and demonstrate its improvements over the original method. The experiments aim to investigate how beam scores, computed at each step of the beam search algorithm during the caption generation process, influence the final output.

5.1.1 Data Preprocessing

Since the algorithm is data-driven, the training data is leveraged in order to extract key measures that can guide the beam search algorithm toward creating improved diagnoses. The preprocessing phase -before generating diagnoses- involved creating all possible pairs of two tags from the available tags of the training dataset. From a set of 2,125 distinct tags (e.g., "Heart", "X-ray Computed Tomography"), all possible tuples were created, resulting in 106,653 unique tuples. For each tuple, we examined the captions associated with both tags to calculate two MMCS scores: one for each tag in relation to these specific captions. As described in Sec. 3, these scores represent how explicitly a tag is represented in these training captions, which are associated with the tuple it belongs to. Using these MMCS scores, the algorithm computes the difference of MCS and MMCS for each tag at every step of the generation. This difference is used to compute the loss, known as the TDMM score, which guides the beam search algorithm to prioritize generated captions that express the biomedical tags in a way consistent with how they are represented in the training captions.

5.1.2 Algorithm Variants

Three slightly different versions of the Tuples-DMM algorithm were tested for the caption generation procedure, with the aim of identifying the most effective approach. All of these versions describe the application of the Tuples-DMM method whenever an image contained more than one tag, while still using the original DMMCS method in cases of

images associated with only one tag. The first version included the straightforward creation of tuples. In this approach, no additional filtering was applied and all generated tuples were considered during the caption generation process. The second one introduced a filtering mechanism that retains only tuples of two tags with maximal contextual dissimilarity, excluding those composed of highly similar tags. The third version implemented the removal of frequent tags in order to form tuples using only infrequent tags explained in Section 3.2.4, followed by the application of the dissimilarity-based filtering mechanism, as described in the second version. Among these versions, the second achieved the highest metric scores. The tuples-filtering process that decides the final number of tuples used was determined through manual experimentation. For example, when an image was associated with three tags, two were retained; for seven tags, five were kept. This selective approach focused on dissimilar tags, as their common associated captions were found to be more informative and relevant.

Here is the pseudocode for the second version of the algorithm that shows how the selection process of the tuples was conducted:

Algorithm 1 Tuple Filtering Based on Tag Dissimilarity

Require: `ranked_tuples` (a dictionary of ranked tuples)
Ensure: `filtered_tuples` (a dictionary with filtered tuples)

```
if len(ranked_tuples) == 3 then
    filtered_tuples ← dict(list(ranked_tuples.items())[:2])
    {Retain top 2 tuples for 3 tags}
else if len(ranked_tuples) == 4 then
    filtered_tuples ← dict(list(ranked_tuples.items())[:3])
    {Retain top 3 tuples for 4 tags}
else if len(ranked_tuples) <= 6 then
    filtered_tuples ← dict(list(ranked_tuples.items())[:4])
    {Retain top 4 tuples for 5-6 tags}
else if len(ranked_tuples) <= 8 then
    filtered_tuples ← dict(list(ranked_tuples.items())[:5])
    {Retain top 5 tuples for 7-8 tags}
else if len(ranked_tuples) >= 9 then
    filtered_tuples ← dict(list(ranked_tuples.items())[:6])
    {Retain top 6 tuples for 9 or more tags}
else
    filtered_tuples ← ranked_tuples {No filtering for other cases}
end if
return filtered_tuples =0
```

To address the third version, we experimented with creating tuples only from infrequent tags. The tags not being considered in the tuples' creation process were: "Anterior-Posterior", "Plain x-ray", "Chest", "Abdomen", "Plain chest X-ray", "Upper Extremity", "Postero-Anterior", "Sagittal", "X-Ray Computed Tomography", "Radiographic imaging procedure", "angiogram".

5.1.3 Experimental Results

The experiments used the same instruction prompt as the baseline to ensure a fair comparison and determine whether the proposed method truly improves on the original. For the Tuples-DMM method, Flan-T5-xl was selected as the LLM and employed with the instruction prompt using 32-bit numerical precision. The instruction prompt used was:

> "You are an experienced radiologist. You are being given radiology images along with a short medical diagnosis. Generate a descriptive caption that highlights the location, nature, and severity of the abnormality of the radiology image."

In the ImageCLEF 2023 biomedical dataset, the evaluation of generated captions emphasized two key metrics: **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) [Lin04] and **BERTScore** [Zha+19] as determined by Rückert et al. [Rüc+23]. The primary focus was on BERTScore because it assesses the semantic similarity by using contextual embeddings, thus capturing context that the traditional methods are unable to. The secondary metric was ROUGE as it measures the overlap of n-grams between generated and ground truth texts, complementing BERTScore's focus on semantics. The focus on these two metrics aims to ensure that the generated descriptions are both lexically and semantically aligned with the ground truth captions. BLEU (BiLingual Evaluation Understudy) [Pap+02], on the other hand, relies on n-gram precision and does not fully capture the meaning of the text. Unlike ROUGE, which rewards the inclusion of key phrases in ground truth texts, BLEU focuses on how much of the generated text exactly matches the reference, often penalizing informative content that is phrased differently. This makes BLEU less suitable than ROUGE for tasks such as captioning, where capturing the core ideas is prioritized over exact wording. Additionally, the BLEURT (Bilingual Evaluation Understudy with Representations from Transformers) [SDP20] metric was introduced due to its ability to align closely with human judgments, but was not considered as important as BERTScore and ROUGE.

Table 5.1 presents the performance metrics for the Tuples-DMM method using filtering based on tags' dissimilarity (Tuples-DMM with Dissimilarity Filtering), by generating diagnostic text for 2,000 biomedical images. Experiments were conducted using different values of the weighting factor α to demonstrate its influence on the model performance. These experimental results compare the performance of the Tuples-DMM and DMM approaches across the evaluation metrics. Tuples-DMM consistently outperforms DMM in the prioritized metrics for ImageCLEF 2023, **ROUGE** and **BERTScore**. Notably:

- **ROUGE:** Tuples-DMM achieved the highest ROUGE score of **20.92** at $\alpha = 0.20$, demonstrating its ability to produce text better aligned with the ground truth diagnoses.

- **BERTScore:** Tuples-DMM also achieved its best BERTScore of **62.26** at $\alpha = 0.20$, indicating improved contextual accuracy in the generated descriptions.

α	Method	ROUGE	BLEURT	BLEU	BERTScore
0.05	Tuples-DMM with DF	20.63	29.10	11.90	62.11
	DMM	**20.79**	**29.26**	**12.30**	**62.23**
0.10	Tuples-DMM with DF	20.74	29.19	12.20	**62.19**
	DMM	**20.93**	**29.51**	**12.90**	62.06
0.15	Tuples-DMM with DF	**20.81**	29.32	12.40	**62.24**
	DMM	20.67	**29.78**	**13.90**	61.47
0.18	Tuples-DMM with DF	**20.88**	29.41	12.60	**62.25**
	DMM	20.54	**29.79**	**14.40**	61.31
0.19	Tuples-DMM with DF	**20.86**	29.41	12.60	**62.25**
	DMM	20.51	**29.83**	**14.60**	61.21
0.20	Tuples-DMM with DF	**20.92**	29.41	12.60	**62.26**
	DMM	20.58	**29.81**	**14.90**	61.19
0.25	Tuples-DMM with DF	**20.89**	29.47	12.90	**62.19**
	DMM	20.43	**29.87**	**15.50**	60.86
0.30	Tuples-DMM with DF	**20.72**	29.50	12.80	**62.20**
	DMM	20.09	**30.02**	**16.20**	60.30

Tab. 5.1: Performance metrics for both Tuples-DMM and DMM experiments evaluated across α thresholds ranging from 0.05 to 0.30.

Although the baseline DMM method performed better in BLEU scores at values of $\alpha > 0.10$, this metric is less significant for this dataset compared to the previous ones. BLEURT scores showed competitive results for both approaches, but DMM remained slightly ahead, even at $\alpha = 0.20$, where Tuples-DMM achieved its best performance in the primary metrics. Overall, $\alpha = 0.20$ emerged as the optimal parameter for Tuples-DMM applying the dissimilarity-based filtering mechanism, as it maximized performance in both ROUGE and BERTScore. These results demonstrate that Tuples-DMM is better suited to meet the dataset's requirements compared to the baseline DMM method.

The subsequent results aim to evaluate the dissimilarity-based filtering mechanism's contribution to the Tuples-DMM approach. For this purpose, we compared the second version of Tuples-DMM with DF to the first version (which involved only tuple creation) using α values of 0.15, 0.20, and 0.25.

For $\alpha = 0.15$, Tuples-DMM with dissimilarity filtering yields better scores across all metrics compared to the approach of creating tuples without filtering. For $\alpha = 0.20$, the dissimilarity-filtering-based method performs better in BERTScore and BLEURT, but

α	Method	ROUGE	BLEURT	BLEU	BERTScore
0.15	Tuples-DMM with DF	**20.81**	**29.32**	12.40	**62.24**
	Tuples-DMM	20.79	29.29	12.40	62.23
0.20	Tuples-DMM with DF	20.92	**29.41**	**12.60**	**62.26**
	Tuples-DMM	**20.93**	29.37	12.60	62.25
0.25	Tuples-DMM with DF	20.89	29.47	12.90	62.19
	Tuples-DMM	**20.91**	29.47	12.90	62.19

Tab. 5.2: Comparison of Tuples Only and Tuples-DMM (with dissimilarity filtering) at Different Thresholds.

slightly worse in ROUGE, while, for $\alpha = 0.25$, it only improves the BERTScore. Since BERTScore is the primary metric, this highlights the significance of dissimilarity filtering in Tuples-DMM.

The other functionality that was tested is the extraction of frequent tags when creating tuples (Tuples-DMM with DF no Frequent Tags). This addition to the method produced the following results in comparison to Tuples-DMM with dissimilarity filtering.

α	Method	ROUGE	BLEURT	BLEU	BERTScore
0.15	Tuples-DMM with DF	**20.81**	29.32	12.40	**62.24**
	Tuples-DMM with DF no FT	20.63	**29.79**	**13.70**	61.49
0.20	Tuples-DMM with DF	**20.92**	**29.41**	12.60	**62.26**
	Tuples-DMM with DF no FT	20.60	**29.85**	**14.60**	61.30
0.25	Tuples-DMM with DF	**20.89**	29.47	12.90	**62.19**
	Tuples-DMM with DF no FT	20.53	**29.89**	**15.10**	61.10

Tab. 5.3: Comparison of Tuples-DMM with Dissimilarity Filtering and Tuples-DMM excluding Frequent Tags and Dissimilarity Filtering at Different Thresholds.

It is evident that this approach is not effective. Although it improves on BLEU and BLEURT, it decreases the values of BERTScore and ROUGE, which are the most critical metrics.

5.1.4 Observations and Analysis

The results indicate that the Tuples-DMM method improves performance, particularly in ROUGE and BERTScore, which are critical metrics for evaluating the quality of generated captions. Incorporating tag dissimilarity further enhanced the relevance and detail of the generated captions. On the other hand, focusing on infrequent tags degraded BERTScore and ROUGE. This suggests that the technique of prioritizing infrequent tags is not well suited for improving caption generation when tags are used to guide the algorithm. Additionally, experiments varying the weighting factor α demonstrated its influence on the model performance. Table 5.1 illustrates the effect of increasing α on the evaluation metrics.

These findings highlight that the Tuples-DMM method with dissimilarity filtering (for α = 0.20) is the superior approach to generate medical captions for radiology images provided by the ImageCLEF 2023 dataset.

6 Conclusions and Future Work

6.1 Conclusions

This thesis aims to improve the generation of descriptive captions for biomedical radiological images, such as X-rays, MRIs, and tomographies. The focus is on modifying a guided decoding method by incorporating a retrieval mechanism that selects the most relevant captions to guide the algorithm. The core retrieval idea is to form pairs of tags and retrieve captions in which at least two of these tags co-occur, expecting that such captions provide richer contextual information. This information is considered meaningful because it represents a medical context that must combine two concepts reflected by both tags. During a modified version of the beam search algorithm, beam scores are computed based on these specific captions in order to generate diagnostic captions that effectively assist clinicians.

Specifically, the proposed method builds upon the baseline approach introduced by Kaliosis et al., which implemented a concept-driven guided decoding algorithm. In that baseline, the algorithm utilized training captions associated with tags assigned to the target image. By analyzing how these training captions typically express the corresponding biomedical tags, the algorithm tries to learn specific patterns to be guided towards expressing these tags similarly in the generated caption.

In this work, we experimented with different techniques to improve this methodology. Specifically, we employed the **InstructBLIP model** with a specific instruction prompt and used a modified version of the beam search algorithm to build the final diagnostic caption step by step. The new method incorporates these three key techniques:

- **Creating tuples of tags and retrieving captions where both tags coexist:** The rationale was that these captions would carry more meaningful information by combining two medical concepts.

- **Filtering tuples based on dissimilarity:** The algorithm was designed to prioritize tuples of dissimilar tags, because combining two sufficiently different concepts could provide more precise training captions to guide the generation process.

- **Excluding the most frequent tags during tuple creation:** Frequent tags were excluded as they often represent generic information that could misguide the algorithm. Infrequent tags, on the other hand, usually reflect more specific and meaningful medical terminology, critical for diagnostic captions.

By integrating the first and second techniques, we achieved improvements in the two most important metrics: **ROUGE** and **BERTScore**. However, the third technique did not yield significant benefits.

6.2 Future Work

Future work may include improving the process of retrieving contextually important captions. For example, creating tag pairs with more than two tags could add could improve the retrieval process and have a greater impact on the final results. Additionally, exploring other models besides InstructBLIP or using different text generation algorithms instead of the beam search algorithm could lead to better outcomes. Another promising direction would be to incorporate additional sources of data, such as patient histories or radiology reports, to make the generated captions more precise and clinically relevant.

Bibliography

[And+18] P. Anderson, X. He, C. Buehler, et al. “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2018, pp. 6077–6086.

[BI24] Mateusz Bartosiewicz and Marcin Iwanowski. “The Optimal Choice of the Encoder–Decoder Model Components for Image Captioning”. In: *Information* 15.8 (2024).

[Che+22] Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. *Generating Radiology Reports via Memory-driven Transformer*. 2022. arXiv: 2010.16056 [cs.CL].

[Cho+14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[Cor+20] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. “Meshed-memory transformer for image captioning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10578–10587.

[Dai+23] Wenliang Dai, Junnan Li, Dongxu Li, et al. *Instructblip: Towards general-purpose vision-language models with instruction tuning. ArXiv abs/2305.06500 (2023)*. 2023.

[Dat+20] Sumanth Dathathri, Andrea Madotto, Janice Lan, et al. *Plug and Play Language Models: A Simple Approach to Controlled Text Generation*. 2020. arXiv: 1912.02164 [cs.CL].

[DCI19] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. *Image Captioning with Unseen Objects*. 2019. arXiv: 1908.00047 [cs.CV].

[Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[DT05] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[Far+10] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, et al. "Every Picture Tells a Story: Generating Sentences from Images". In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–29.

[Fer+10] Stefano Ferilli, Marenglen Biba, Teresa M. A. Basile, and Floriana Esposito. "Using Explicit Word Co-occurrences to Improve Term-Based Text Retrieval". In: *Proceedings of the X Conference on Information Retrieval (CIR)*. 2010.

[FMR08] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE conference on computer vision and pattern recognition*. Ieee. 2008, pp. 1–8.

[GPM23] Taraneh Ghandi, Hamidreza Pourreza, and Hamidreza Mahyar. "Deep Learning Approaches on Image Captioning: A Review". In: *ACM Computing Surveys* 56.3 (Oct. 2023), pp. 1–39.

[Guu+20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. *REALM: Retrieval-Augmented Language Model Pre-Training*. 2020. arXiv: 2002.08909 [cs.CL].

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[Hoc97] S Hochreiter. "Long Short-term Memory". In: *Neural Computation MIT-Press* (1997).

[Hos+19] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. "A comprehensive survey of deep learning for image captioning". In: *ACM Computing Surveys (CsUR)* 51.6 (2019), pp. 1–36.

[Jia+18] Wenhao Jiang, Lin Ma, Xiang Chen, Hanwang Zhang, and Wei Liu. "Learning to Guide Decoding for Image Captioning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. AAAI Press, Apr. 2018.

[Jin+20] HanQi Jin, Yue Cao, TianMing Wang, XinYu Xing, and XiaoJun Wan. "Recent advances of neural text generation: Core tasks, datasets, models and challenges". In: *Science China Technological Sciences* 63.10 (2020), pp. 1990–2010.

[Jon+23] Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, et al. *FiDO: Fusion-in-Decoder optimized for stronger performance and faster inference*. 2023. arXiv: 2212.08153 [cs.CL].

[Kal+24] Panagiotis Kaliosis, John Pavlopoulos, Foivos Charalampakos, Georgios Moschovis, and Ion Androutsopoulos. *A Data-Driven Guided Decoding Mechanism for Diagnostic Captioning*. 2024. arXiv: 2406.14164 [cs.AI].

[Kar+20] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, et al. *Dense Passage Retrieval for Open-Domain Question Answering*. 2020. arXiv: 2004.04906 [cs.CL].

[KB13] Nal Kalchbrenner and Phil Blunsom. "Recurrent continuous translation models". In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, pp. 1700–1709.

[Kha+20] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. *Generalization through Memorization: Nearest Neighbor Language Models*. 2020. arXiv: 1911.00172 [cs.CL].

[KK20] Sulabh Katiyar and Samir Kumar. "Comparative Evaluation of CNN Architectures for Image Caption Generation". In: *International Journal of Advanced Computer Science and Applications* 11.12 (2020).

[KLM96] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. "Reinforcement learning: A survey". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285.

[KP06] April Kontostathis and William M Pottenger. "A framework for understanding Latent Semantic Indexing (LSI) performance". In: *Information Processing & Management* 42.1 (2006), pp. 56–73.

[KPA19] Vasiliki Kougia, John Pavlopoulos, and Ion Androutsopoulos. *A Survey on Biomedical Image Captioning*. 2019. arXiv: 1905.13302 [cs.CV].

[Kra+20] Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, et al. *GeDi: Generative Discriminator Guided Sequence Generation*. 2020. arXiv: 2009.06367 [cs.CL].

[Lew+21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. arXiv: 2005.11401 [cs.CL].

[Li+22] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. *A Survey on Retrieval-Augmented Text Generation*. 2022. arXiv: 2202.01110 [cs.CL].

[Lin04] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[Low99] David G Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[LXW19] Xiaoxiao Liu, Qingyang Xu, and Ning Wang. "A survey on deep neural network-based image captioning". In: *The Visual Computer* 35.3 (Mar. 2019), pp. 445–470.

[MMS00] Elke Mittendorf, Bojidar Mateev, and Peter Schäuble. "Using the Co-occurrence of Words for Retrieval Weighting". In: *Journal of Information Retrieval* (June 2000).

[Nat23] National Library of Medicine. *UMLS Knowledge Sources*. [Accessed: 2024-12-19]. 2023.

[OPM02] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.7 (2002), pp. 971–987.

[OT01] Aude Oliva and Antonio Torralba. "Modeling the shape of the scene: A holistic representation of the spatial envelope". In: *International journal of computer vision* 42 (2001), pp. 145–175.

[Pan+20] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. "X-linear attention networks for image captioning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10971–10980.

[Pap+02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[Par+21a] H. Park, K. Kim, S. Park, and J. Choi. "Medical Image Captioning Model to Convey More Details: Methodological Comparison of Feature Difference Generation". In: *IEEE Access* 9 (2021), pp. 150560–150568.

[Par+21b] Hyeryun Park, Kyungmo Kim, Seongkeun Park, and Jinwook Choi. "Medical Image Captioning Model to Convey More Details: Methodological Comparison of Feature Difference Generation". In: *IEEE Access* 9 (2021), pp. 150560–150568.

[Poe+22] Gabriel Poesia, Oleksandr Polozov, Vu Le, et al. *Synchromesh: Reliable code generation from pre-trained language models*. 2022. arXiv: 2201.11227 [cs.LG].

[RA23] Abir Rahali and Moulay A Akhloufi. "End-to-end transformer-based models in textual-based NLP". In: *AI* 4.1 (2023), pp. 54–110.

[Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving Language Understanding by Generative Pre-Training". In: *OpenAI* (2018).

[Ras+24] Ahmad Rashid, Ruotian Wu, Julia Grosse, Agustinus Kristiadi, and Pascal Poupart. *A Critical Look At Tokenwise Reward-Guided Text Generation*. 2024. arXiv: 2406.07780 [cs.LG].

[RJ76] Stephen E Robertson and K Sparck Jones. "Relevance weighting of search terms". In: *Journal of the American Society for Information science* 27.3 (1976), pp. 129–146.

[Rüc+23] Johannes Rückert, Asma Ben Abacha, Alba G Seco de Herrera, et al. "Overview of ImageCLEFmedical 2023–caption prediction and concept detection". In: *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023)*. Vol. 3497. 108. 2023, pp. 1328–1346.

[Sal83] Gerard Salton. "Modern information retrieval". In: *(No Title)* (1983).

[SDP20] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. "BLEURT: Learning robust metrics for text generation". In: *arXiv preprint arXiv:2004.04696* (2020).

[Shi+16] Hoo-Chang Shin, Kirk Roberts, Le Lu, et al. "Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[SJS22] K. Revati Suresh, Arun Jarapala, and P. V. Sudeep. "Image Captioning Encoder–Decoder Models Using CNN–RNN Architectures: A Comparative Study". In: *Circuits, Systems, and Signal Processing* 41.10 (Oct. 2022), pp. 5719–5742.

[SMH11] Ilya Sutskever, James Martens, and Geoffrey E Hinton. "Generating text with recurrent neural networks". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 1017–1024.

[Sne24] Eriks Sneiders. "Text Retrieval by Term Co-occurrences in a Query-based Vector Space". In: *Conference on Information Retrieval and Applications (CIRA)*. Department of Computer and Systems Sciences, Stockholm University. 2024.

[SP23] Himanshu Sharma and Devanand Padha. "A comprehensive survey on image captioning: from handcrafted to deep learning-based techniques, a taxonomy and open research issues". In: *Artificial Intelligence Review* 56.11 (2023), pp. 13619–13661.

[Ste+23] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, et al. "From Show to Tell: A Survey on Deep Learning-Based Image Captioning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2023), pp. 539–559.

[TBH21] M. Onat Topal, Anil Bas, and Imke van Heerden. *Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet*. 2021. arXiv: 2102.08036 [cs.CL].

[Vas17] A Vaswani. "Attention is all you need". In: *Advances in Neural Information Processing Systems* (2017).

[Vin+17] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.4 (2017), pp. 652–663.

[VW97] Robert A Van De Geijn and Jerrell Watts. "SUMMA: Scalable universal matrix multiplication algorithm". In: *Concurrency: Practice and Experience* 9.4 (1997), pp. 255–274.

[Wan+19] Yiyu Wang, Jungang Xu, Yingfei Sun, and Ben He. *Image Captioning based on Deep Learning Methods: A Survey*. 2019. arXiv: 1905.08110 [cs.CV].

[WC16] ZYYYY Wu and RSWW Cohen. "Encode, review, and decode: Reviewer module for caption generation". In: *arXiv preprint arXiv:1605.07912* 3 (2016).

[Wu+16] Yonghui Wu, Mike Schuster, Zhifeng Chen, et al. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. 2016. arXiv: 1609.08144 [cs.CL].

[WXS22] Yiyu Wang, Jungang Xu, and Yingfei Sun. *End-to-End Transformer Based Model for Image Captioning*. 2022. arXiv: 2203.15350 [cs.CV].

[XDY19] Yuxuan Xiong, Bo Du, and Pingkun Yan. "Reinforced transformer for medical image captioning". In: *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*. Springer. 2019, pp. 673–680.

[Xu+15] Kelvin Xu, Jimmy Ba, Ryan Kiros, et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2048–2057.

[Xu+16] Kelvin Xu, Jimmy Ba, Ryan Kiros, et al. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. 2016. arXiv: 1502.03044 [cs.LG].

[Yan+20] Zhilin Yang, Zihang Dai, Yiming Yang, et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. 2020. arXiv: 1906.08237 [cs.CL].

[YK21] Kevin Yang and Dan Klein. "FUDGE: Controlled Text Generation With Future Discriminators". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021.

[Zha+19] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. "Bertscore: Evaluating text generation with bert". In: *arXiv preprint arXiv:1904.09675* (2019).

[Zha+23] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. "A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models". In: *ACM Comput. Surv.* 56.3 (Oct. 2023).

[ZK22] Zanyar Zohourianshahzadi and Jugal K. Kalita. "Neural attention for image captioning: review of outstanding methods". In: *Artificial Intelligence Review* 55.5 (June 2022), pp. 3833–3862.

[ZVS21] Sina Zarrieß, Henrik Voigt, and Simeon Schüz. "Decoding Methods in Neural Language Generation: A Survey". In: *Information* 12.9 (2021), p. 355.

List of Acronyms

MRI Magnetic Resonance Imaging

CT Computed Tomography

DC Diagnostic Captioning

CV Computer Vision

NLP Natural Language Processing

AI Artificial Intelligence

LBP Local Binary Patterns

GIST Global Image Descriptor

SIFT Scale-Invariant Feature Transform

HOG Histograms of Oriented Gradients

SVM Support Vector Machine

LSTM Long Short-Term Memory

GRU Gated Recurrent Unit

GPT Generative Pre-trained Transformer

BERT Bidirectional Encoder Representations from Transformers

NLG Natural Language Generation

CTG Controllable Text Generation

IPO Input-Process-Output

RL Reinforcement Learning

RGTG Reward-Guided Text Generation

PPLM Plug and Play Language Models

SuMMa Successive Multiplication of Matrices

LSE Latent Semantic with Explicit co-occurrences

ROUGE Recall-Oriented Understudy for Gisting Evaluation

BLEU BiLingual Evaluation Understudy

BLEURT Bilingual Evaluation Understudy with Representations from Transformers

CNN Convolutional Neural Networks

RNN Recurrent Neural Networks

DF Dissimilarity Filtering

FT Frequent Tags

List of Figures

List of Tables

List of Algorithms